Recommender Systems for the Conference Paper Assignment Problem

Don Conry[†], Yehuda Koren[$], and Naren Ramakrishnan[†]
[†]Department of Computer Science, Virginia Tech, VA 24061, USA
[$]Yahoo! Research, Israel

ABSTRACT

Conference paper assignment, i.e., the task of assigning paper submissions to reviewers, presents multi-faceted issues for recommender systems research. Besides the traditional goal of predicting 'who likes what?', a conference management system must take into account aspects such as: reviewer capacity constraints, adequate numbers of reviews for papers, expertise modeling, conflicts of interest, and an overall distribution of assignments that balances reviewer preferences with conference objectives. Among these, issues of modeling preferences and tastes in reviewing have traditionally been studied separately from the optimization of paper-reviewer assignment. In this paper, we present an integrated study of both these aspects. First, due to the paucity of data per reviewer or per paper (relative to other recommender systems applications) we show how we can integrate multiple sources of information to learn paper-reviewer preference models. Second, our models are evaluated not just in terms of prediction accuracy but in terms of the end-assignment quality. Using a linear programming-based assignment optimization formulation, we show how our approach better explores the space of unsupplied assignments to maximize the overall affinities of papers assigned to reviewers. We demonstrate our results on real reviewer preference data from the IEEE ICDM 2007 conference.

Categories and Subject Descriptors

H.4.2 [**Information Systems Applications**]: Types of Systems—*Decision support*; J.4 [**Computer Applications**]: Social and Behavioral Sciences

Keywords

Recommender systems, collaborative filtering, conference paper assignment, linear programming.

1. INTRODUCTION

Modern conferences, especially in areas such as data mining/machine learning (KDD; ICDM; ICML; NIPS) and databases/web (VLDB; SIGMOD; WWW), are beset with excessively high numbers of paper submissions. Assigning these papers to appropriate reviewers in the program committee (which can constitute a few hundred members) is a herculean task and hence motivates the use of recommender systems.

Besides the traditional goal of predicting 'who likes what?', a conference management system must take into account aspects such as: reviewer capacity constraints, adequate numbers of reviews for papers, expertise modeling, conflicts of interest, and an overall distribution of assignments that balances reviewer preferences with conference objectives. Among these, issues of modeling preferences, expertise, and tastes in reviewing have traditionally been studied separately from the optimization of paper-reviewer assignment. The former has been the subject of much academic research (see Section 2.1) while the latter is emphasized by commercial software, such as EasyChair, CyberChair, and Microsoft's CMS, which aim to automate the management of the conference reviewing process.

We investigate the conference paper assignment problem (CPAP) through the lens of recommender systems research. There are three key differences from traditional recommender systems research and the CPAP problem. First, in a traditional recommender, recommendations that meet the needs of one user do not affect the satisfaction of other users. In CPAP, on the other hand, multiple users (reviewers) are bidding to review the same papers and hence there is the possibility of one user's recommendations (assignments) affecting the satisfaction levels (negatively) of other users. Hence the design of reviewer preference models must be posed and studied in an overall optimization framework.

Second, in a conventional recommender, the goal is often to recommend *new* entities that are likely to be of interest, whereas in CPAP, the goal is to ensure that reviewers are predominantly assigned their (most) preferred papers. Nevertheless, preference modeling is still crucial because it gives the assignment algorithm some degree of latitude in aiming to satisfy multiple users.

Finally, recommender systems are used to working with sparse data but the amount of 'signal' available to model preferences in the CPAP domain is exceedingly small; hence we must integrate multiple sources of information to build strong preference models.

In this paper, we present the first integrated study of both modeling reviewing preferences and optimizing assignments for conference management. Our key contributions can be summarized as follows.

1. Due to the paucity of data per reviewer or per paper (relative to other recommender systems applications) we show how we can integrate information about publication subject categories, contents of paper abstracts, and co-authorship information to learn improved paper-reviewer preference models.

2. We evaluate our models not just in terms of prediction accuracy but in terms of the end-assignment quality. Using a linear programming-based assignment optimization formulation, we show how our approach bet-

ter explores the space of unsupplied assignments to maximize the overall affinities of papers assigned to reviewers.

3. We demonstrate the effectiveness of our approach on actual reviewing preference data in the context of a real life conference, namely the IEEE ICDM'07 conference [19].

2. RELATED RESEARCH

Any conference management system must contend with two main issues: how to model affinities or preferences between papers and reviewers, and how to use these affinities to make and/or optimize assignments. For the former issue, many conferences have an explicit 'bidding' phase and use data collected in this phase as the affinity matrix. While many conferences use these bids as-is, we will demonstrate how they can be used as the starting point to build improved preference models. Approaches to solve the latter issue have traditionally been considered orthogonal to the problem of preference modeling but, as we demonstrate later, better preference modeling leads to improvements in this phase as well.

2.1 Modeling Affinities, Preferences, and Expertise

The sparsity of reviewer-paper bidding data has led some researchers, e.g., Rigaux [20], to explore the use of collaborative filtering techniques [6, 11] to 'grow' the given bids. The underlying assumption is that reviewers who bid similarly on a number of the same papers are likely to have similar preferences for other papers. Basu et al [1] use the relational WHIRL system to integrate similarity scores from disparate data sources to identify most relevant (paper,reviewer) combinations. They do not, however, attempt to satisfy per-paper or per-reviewer constraints, and the contributions of different sources are considered equivalent to each other. Popescul et al [18] present a way to combine content-based and collaborative recommendations using a three-way aspect model. The GRAPE system [14] prefers topical information over supplied reviewer bids or preferences, but does use the preferences as a secondary means of modeling. The rationale is the view that topical data more accurately predicts the degree of expertise present for a reviewer-paper match. Since the distribution of reviewers and papers over topics is unpredictable (sometimes leaving too many or too few reviewers for a given cluster of papers), the preference information is used for tuning or smoothing out the wrinkles in the topic-based assignments.

A problem faced by most expertise modeling approaches is identifying which topics are covered in papers. Early efforts in this area focused mainly on paper abstracts, and topical expertise was determined through common information retrieval methods involving keywords. For example, Dumais and Nelson [5] match papers to reviewers using Latent Semantic Indexing (LSI) trained on reviewer-supplied abstracts. Yarowsky and Florian [27] extended this idea by using a similar vector space model with a naive Bayes classifier on work previously published by each reviewer.

More recently, Wei & Croft [26] describe a topic-based model using a language model with Dirichlet smoothing. An excellent example of topic-based models is the Author-Persona-Topic (APT) model by Mimno & McCallum [16]. The APT model contains a number of features designed to better capture the reality of the relationship between conference reviewers and papers. The idea is that an author may study and write about several distinct topics; by clustering papers from each of these topics into a separate persona for an author, the author's ranking for a given topic need not be diluted by his or her writings on a different topic.

2.2 Optimizing Assignments

Given preference data, either explicitly gathered or computationally modeled, the actual task of making assignments can be viewed as bipartite matching. The classical approach to bipartite matching is given by the Hungarian Algorithm described by Kuhn [13]; it provides a solution for the simplest cases of this family of problems (applicable when the number of reviewers equals the number of papers). Various refinements have been made to this algorithm over the years, such as one by Hopcroft and Karp [10]. A number of contemporary assignment systems take this approach, including GRAPE [14]. For practical reasons, it is useful to restrict the number of reviews per reviewer and per paper; a constraint based linear program, e.g., work by Taylor [25], is a natural approach.

Another approach to CPAP uses reasoning from the much more general *minimal cost network flow* problems studied in dynamics and operations research. Many such related problems (known collectively as extended Generalized Assignment Problems [2] or GAP) of assigning a limited number of resources to certain tasks exist in diverse fields. In the network flow diagram of this general assignment problem, resources (in our case, reviewers) are represented by source nodes with a certain supply (number of reviews allowed per reviewer), while tasks (each paper to be reviewed) are sink nodes with a demand (the number of times each paper must be reviewed). For specific approaches, see [7, 8].

3. MODELS OF REVIEW PREFERENCES

We adapt recommendation techniques for predicting unknown reviewer-paper preferences. Naturally, reviewers assume the role of "users" in traditional recommender systems, while papers take the role reserved to "products". Our goal is to exploit a variety of available information (see Fig. 1) in order to get better estimates of those unknown preferences. This, in turn, will allow the assignment algorithm to find better matches between reviewers and papers. First, we introduce some essential conventions.

3.1 Notation and Dataset Description

We are given *ratings* (henceforth, interchangeable with *preferences*) about m reviewers and n papers. We reserve special indexing letters for distinguishing reviewers from papers: for reviewers u, v, and for papers i, j. A rating r_{ui} indicates the preference by reviewer u of paper i, where high values mean stronger preferences. Usually the vast majority of ratings are unknown.

As a concrete example, the dataset utilized in this paper comes from the Seventh IEEE International Conference on Data Mining (ICDM'07) held in Omaha, NE, USA (utilized here with permission). The originally supplied matrix is sparse: 529 papers, 203 reviewers, and only 6267 bids. This means that a reviewer rates about 31 papers on average, while a paper recieves less than 12 ratings on average. Each rating reflects a bid a reviewer put on a paper, with

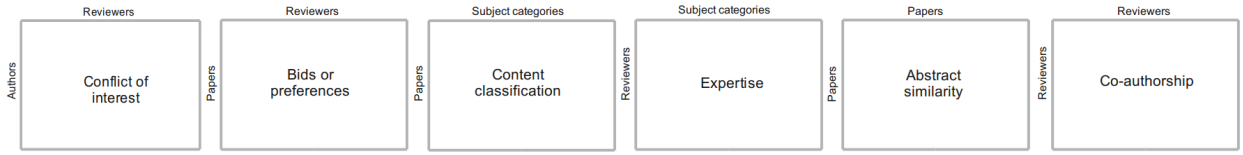

Figure 1: Data used in this paper for building paper-reviewer preference models.

numerical values, between 1 and 4, indicating preferences as follows: 4= "High", 3="OK", 2="Low" and 1="No".

We distinguish predicted ratings from known ones, by using the notation $\hat{r}_{ui}$ for the predicted value of r_{ui}. To evaluate the models we split the dataset into a train set, which contains about 90% of the preferences (randomly chosen), and a test set, which contains the rest preferences. Consequently, our models learn the train set and assign values to $\hat{r}_{ui}$ for all (u, i)-pairs in the test set. Results from these runs are averaged over 100 iterations of training-test data splits.

The quality of the results on a specific test set ($TestSet$) is measured by their root mean squared error (RMSE): $\sqrt{\sum_{(u,i)\in TestSet}(r_{ui} - \hat{r}_{ui})^2/|TestSet|}$. The overall accuracy of the model is taken as the mean RMSE over the 100 randomly generated test sets. The reason for using such a randomization is the small size of our dataset, which makes each individual test set relatively small.

We hasten to add that we do not advocate the myopic view of RMSE [15] as the primary criterion for recommender systems evaluation. We use it in this section primarily due to its convenience for constructing direct optimizers. In the next section we will evaluate performance according to criteria more natural to the paper assignment problem. We also note that small improvements in overall RMSE will typically translate into substantial improvements in bottom-line performance for predicting paper-reviewer preferences.

In the following, we gradually expand the prediction model, by introducing into it a growing set of features.

3.2 Baseline model

Much of the variability in the data is explained by global effects, which can be reviewer- or paper-specific. It is important to capture this variability by a separate component, thus letting the more involved models deal only with genuine reviewer-paper interactions. We model these global effects through:

$$\hat{r}_{ui} = \mu + b_u + b_i \tag{1}$$

The constant μ indicates a global bias in the data, which is taken to be the overall mean rating. The parameter b_u captures reviewer-specific bias, accounting for the fact that different reviewers use different rating scales. Finally, the paper bias, b_i, accounts for the fact that certain papers tend to attract higher (or, lower) bids than others.

We learn optimal values for b_u $(u = 1, \ldots, m)$ and b_i $(i = 1, \ldots, n)$, by minimizing the associated squared error function (or, equivalently, the train RMSE):

$$\min_{b_*} \sum_{(u,i)\in TrainSet} (r_{ui} - \mu - b_u - b_i)^2 + \lambda_1 b_u^2 + \lambda_2 b_i^2$$

The regularizing term, i.e., $\lambda_1 b_u^2 + \lambda_2 b_i^2$, avoids overfitting by penalizing the magnitudes of the parameters. We set the values of the constants λ_1 and λ_2 by cross validation. Learning is done by stochastic gradient descent (alternatively, any least squares solver could be used here). The resulting average test RMSE is **0.6286**.

A separate analysis of each of the two biases shows reviewer effect ($\mu + b_u$, with RMSE **0.6336**) to be much more significant than paper bias ($\mu + b_i$, RMSE **1.2943**) in reducing the error. This indicates a tendency of reviewers to concentrate all ratings near their mean ratings, which is supported by examination of the data.

While the baseline model could explain much of the data variability, as evident by its relatively low associated RMSE, it is useless for making actual assignments. After all, it gives all reviewers exactly the same order of paper preferences. Thus, we are really after the remaining unexplained variability, where reviewer-specific preferences are getting expressed. Uncovering these preferences is the subject of the next subsections.

3.3 A factor model

Latent factor models comprise a common approach to collaborative filtering with the goal to uncover latent features that explain observed ratings; examples include pLSA [9], neural networks [22], and Latent Dirichlet Allocation [3]. We will focus on models that are induced by factorization of the reviewer-paper ratings matrix, which recently have gained popularity [4, 12, 17, 21, 24], thanks to their attractive accuracy and scalability.

The premise of such models is that both reviewers and papers can be characterized as vectors in a common f-D space. The interaction between reviewers and papers is modeled by inner products in that space. Together, with the non-interaction signal covered in the previous subsection, a rating is predicted by the rule:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i \tag{2}$$

Here, $p_u \in \mathbb{R}^f$ and $q_i \in \mathbb{R}^f$ are the factor vectors of reviewer u and paper i, respectively. These are learnt by minimizing the associated squared error function, using stochastic gradient descent. The resulting average test RMSE is slowly decreasing when increasing the dimensionality of the latent factor space. E.g., for $f = 50$ it is **0.6240**, and for $f = 100$ it is **0.6234**. Henceforth, we use $f = 100$.

3.4 Subject categories

While latent factor models automatically infer suitable categories, much can be learnt by known categories attributed to both papers and reviewers. In a typical conference submission process, authors are requested to denote primary and secondary categories appropriate for their papers. Likewise, reviewers are asked to indicate their interest along the same categories. It would be desirable to match reviewers with papers lying within their area of expertise.

More specifically, for our dataset, which contains a number of predefined categories judged relevant for ICDM'07 (see Table 1), the entered matching between paper i and category c is denoted by:

$$\sigma_{ic} = \begin{cases} 1 & c \in \text{primary}(i) \\ \frac{1}{2} & c \in \text{secondary}(i) \\ 0 & \text{otherwise} \end{cases}$$

The value assignment (1 for "primary", 0.5 for "secondary") is derived by cross validation and is quite intuitive. Similarly, we use the following for matching reviewers with their desired categories:

$$\theta_{uc} = \begin{cases} 1 & c \in \text{interest}(u) \\ -\frac{1}{2} & c \in \text{conflict}(u) \\ 0 & \text{otherwise} \end{cases}$$

Notice that in our dataset, reviewers could enter negative interest in certain categories, with which they have a "conflict of interest".

This leads to a model, which measures the interaction between reviewers and papers based on the association of the respective entered categories, leading to:

$$\hat{r}_{ui} = \mu + b_u + b_i + \sum_c \sigma_{ic}\theta_{uc}w_c \tag{3}$$

The weights w_c indicate the significance of each category in linking a reviewer to a paper. Those are learnt automatically from the data by minimizing the squared error on the train set. It is plausible that, e.g., a mutual interest in some category A, will strongly link a reviewer to a paper, while a mutual interest in another category B is less influential on papers choice. For a concrete example, refer to Table 1, which shows the categories in our dataset sorted by their respective w_c values. We observe differences of orders of magnitude in the ability of different categories to correctly predict associations of reviewers to papers. Note in particular that there is no obvious monotonic relationship between the weight imputed to categories and the number of papers/reviewers associated with the category.

The resulting average test RMSE of the model is: **0.6243**. This can be improved by integrating with the latent factor model, yielding:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i + \sum_c \sigma_{ic}\theta_{uc}w_c \tag{4}$$

The RMSE here is **0.6197**.

3.5 Paper-paper similarities

We inject paper-paper similarities into our models in a way reminiscent of item-item recommenders [23]. The building blocks here are similarity values s_{ij}, which measure the similarity of paper i and paper j. The similarities could be derived from the ratings data, but those are already covered by the latent factor model. Rather, we derive the similarity of two papers by computing the cosine of their abstracts. Usually we work with the square of the cosine, which better contrasts the higher similarities against the lower ones.

This leads to a model where a reviewer's preference for a paper is derived from his preferences to similar papers, through a weighted average, as follows:

$$\hat{r}_{ui} = \mu + b_u + b_i + \gamma \frac{\sum_{j \in \mathrm{R}(u)} s_{ij} r_{uj}}{\alpha + \sum_{j \in \mathrm{R}(u)} s_{ij}} \tag{5}$$

Here, the set $\mathrm{R}(u)$ contains all papers on which u bid. The constant α is for regularization: it is penalizing cases where the weighted average has very low support, that is $\sum_{j \in \mathrm{R}(u)} s_{ij}$ is very small (e.g., no similar paper was rated by u). In our dataset it was determined by cross validation to be 0.001. The parameter γ sets the overall weight of the paper-paper component. It is learnt as part of the optimization process (cross-validation could have been used as well). Its final value is closely 0.7. Overall, the resulting average test RMSE of this model is **0.6109**, which is better than what other models could achieve so far.

As usual, we combine the paper-paper similarities into our overall scheme, which further drops RMSE to: **0.6038**, through the following model:

$$\hat{r}_{ui} = \mu + b_u + b_i + p_u^T q_i + \sum_c \sigma_{ic}\theta_{uc}w_c + \gamma \frac{\sum_{j \in \mathrm{R}(u)} s_{ij} r_{uj}}{\alpha + \sum_{j \in \mathrm{R}(u)} s_{ij}} \tag{6}$$

3.6 Reviewer-reviewer similarities

In analogy to paper-paper similarities, one can also use reviewer-reviewer similarities, in order to borrow preferences between like minded reviewers. This is reminiscent of classic user-user collaborative filtering. Once again, we do not want to derive user-user relations directly from their preferences, as the signal from there is already incorporated into the latent factor model. Instead, we resort to an additional data source for deriving those similarities. Here, one can use the publication histories of the reviewers. To model reviewer-reviewer similarities, we utilize the number of commonly co-authored papers as reported in DBLP, denoted by s_{uv}. (More sophisticated choices are of course open for future exploration.) In parallel to the paper-paper model, a preference can be predicted by following the rule:

$$\hat{r}_{ui} = \mu + b_u + b_i + \phi \frac{\sum_{v \in \mathrm{R}(i)} s_{uv} r_{vi}}{\beta + \sum_{v \in \mathrm{R}(i)} s_{uv}} \tag{7}$$

Here, the set $\mathrm{R}(i)$ contains all reviewers that rated i. The regularizing constant β is penalizing cases where the weighted average has very low support, that is, $\sum_{v \in \mathrm{R}(i)} s_{uv}$ is very small (e.g., no similar reviewer has rated i). It was determined by cross validation to be 0.001. The parameter ϕ sets the overall weight of the reviewer-reviewer component. It is learnt as part of the optimization process, with final value close to 0.06 for our dataset. (Notice that ϕ is much smaller than the analogous weight of the paper-paper component, $\gamma = 0.7$.) Overall, the resulting RMSE of this model is **0.6262**, thus offering less accuracy than its dual – the paper-paper model. In other settings, where higher quality reviewer-reviewer similarities are available, the relative merit of the model may increase.

3.7 Putting it all together

The overall model benefits from integrating into it the reviewer-reviewer component by the combined rule:

$$\begin{aligned} \hat{r}_{ui} = & \mu + b_u + b_i + p_u^T q_i + \sum_c \sigma_{ic}\theta_{uc}w_c + \gamma \frac{\sum_{j \in \mathrm{R}(u)} s_{ij} r_{uj}}{\alpha + \sum_{j \in \mathrm{R}(u)} s_{ij}} \\ & + \phi \frac{\sum_{v \in \mathrm{R}(i)} s_{uv} r_{vi}}{\beta + \sum_{v \in \mathrm{R}(i)} s_{uv}} \end{aligned} \tag{8}$$

Table 1: Subject categories used for associating reviewers and papers. Categories are ranked by their weights, which indicate the ability of each category to match papers to appropriate reviewers, as learnt by our model. For comparison the number of papers (assigned to the topic) and reviewers (claiming expertise in the topic) are also shown.

Category	Weight	# reviewers	# papers primary (secondary)
Healthcare, epidemic modeling, and clinical research	0.395121	31	7 (7)
Security, privacy, and data integrity	0.334821	23	12 (6)
Handling imbalanced data	0.284398	24	6 (10)
Data mining in electronic commerce, such as recommendation, sponsored web search, advertising, and marketing tasks	0.260062	39	16 (19)
Mining textual and unstructured	0.245319	66	38 (30)
Intrusion detection, fraud prevention, and surveillance	0.23251	28	7 (12)
Statistical foundations for robust and scalable data mining	0.228847	23	9 (16)
Quality assessment, interestingness analysis, and post-processing	0.21166	30	11 (12)
Mining in networked settings: web, social and computer networks, and online communities	0.206318	62	44 (29)
Mining high speed data streams	0.172367	40	18 (8)
Human-machine interaction and visual data mining	0.168258	23	7 (9)
Telecommunications, network and systems management	0.152845	11	2 (3)
Computational finance, online trading, and analysis of markets	0.11785	18	5 (6)
Bioinformatics, computational chemistry, geoinformatics, and other science engineering disciplines	0.108648	51	14 (26)
Mining sequences and sequential data	0.102578	57	20 (19)
Automating the mining process and other process related issues	0.098819	10	6 (8)
Novel data mining algorithms in traditional areas (such as classification, regression, clustering, probabilistic modeling, and association analysis)	0.089248	91	147 (71)
Mining spatial and temporal datasets	0.081676	45	22 (16)
Customer relationship management	0.081414	21	0 (3)
Mining sensor data	0.05508	40	8 (12)
Dealing with cost sensitive data and loss models	0.03453	12	4 (4)
Data pre-processing, data reduction, feature selection, and feature transformation	0.012069	46	33 (43)
High performance implementations of data mining algorithms	0.008198	38	13 (24)
Algorithms for new, structured, data types, such as arising in chemistry, biology, environment, and other scientific domains	0.006015	60	21 (25)
Distributed data mining and mining multi-agent data	0.000255	29	4 (8)
Developing a unifying theory of data mining	0	36	4 (7)

All parameters are learnt simultaneously by minimizing the associated squared error on the train set. This is our final prediction rule, which delivers an average test RMSE of **0.6015**. In the following section, we will show how filling up the unknown preferences using this model provides flexibility that enables deriving better paper assignments.

4. OPTIMIZING PAPER ASSIGNMENT

Our predicted preference matrix is now suitable for use with any of the optimization algorithms in Section 2.2. Denoting the output of our preference modeling as the affinity matrix $\mathbf{P}$, the assignment problem can be formulated as motivated in Taylor [25]:

$$\begin{aligned} \underset{\mathbf{R}}{\text{argmax}} \quad & \text{trace}\left(\mathbf{P}^T\mathbf{R}\right) = \underset{\mathbf{R}}{\text{argmax}} \sum_u \sum_j \mathbf{P}_{uj}\mathbf{R}_{uj}, \qquad (9) \\ & \text{where} \quad \mathbf{R}_{uj} \in [0,1] \qquad \forall u, j, \\ & \text{and} \quad \sum_j \mathbf{R}_{uj} \leq c_p, \qquad \forall u, \\ & \text{and} \quad \sum_u \mathbf{R}_{uj} \leq c_r, \qquad \forall j. \end{aligned}$$

Here c_p represents the desired number of reviews per paper, and c_r is the desired maximum reviews per reviewer. The third and fourth lines in the equation above represent the constraints on the number of assignments for individual papers and to individual reviewers, respectively. Then the expression trace $\left(\mathbf{P}^T\mathbf{R}\right)$ represents the global sum of affinity, or happiness of all reviewers across all assigned papers. In particular, by using the (binary) assignments matrix $\mathbf{R}$ as a factor, only the affinities from $\mathbf{P}$ for reviewer-paper combinations that exist in the final assignments $\mathbf{R}$ are counted in the sum.

This integer programming problem (9) is reformulated into an easier-to-manage linear programming problem by a series of steps, using the node-edge adjacency matrix, where every row corresponds to a node in $\mathbf{R}$, and every column represents an edge [25]. This reformulation is a bit more complicated, but essentially maps the problem into the domain of linear programming and hence solvable via methods such as Simplex or interior point programming. In particular, as Taylor shows in [25], because the reformulated constraint matrix is *totally unimodular*, there exists at least one globally optimal solution (assignment set) with integral (and due to the constraints, Boolean) coefficients.

5. EXPERIMENTAL RESULTS

We have already demonstrated the ability of our modeling to better capture reviewer-paper preferences. But do the improved models lead to better assignments? In other words, does the assignment algorithm leverage the improved modeling of preferences in ways that improve end-assignment quality? The key distinction is between *preferences* versus *assignments*, an aspect that has not been emphasized in prior recommender systems research.

We study these issues in the context of the IEEE ICDM'07 conference data as described earlier. Data from real conferences is quite rare to come by (e.g., acknowledged also in [16]) and in the future we hope that more datasets will become available to boost recommender systems research in

conference management.

The primary questions we seek to investigate are:

1. Do our preference models lead to improved topical relevance of assignments?
2. Do our preference models lead to higher quality assignments?

We use our preference model (8) to predict ratings for potential assignments for which no expressed preferences exist. Before doing assignments using Taylor's model (9), it is important to balance the rating scale of various reviewers. For example, some reviewers are very cooperative and tend to give mostly high ratings, while others are more cautious and give medium to low ratings. Taylor's model may concentrate only on reviewers with high ratings, which is undesirable. Thus, we suggest two alternative per-reviewer normalization strategies:

1. Subtract the per-reviewer mean from each predicted rating to find the **residual** rating for each potential assignment combination. (Henceforth dubbed as **Resid**.)
2. Calculate **normalized** ratings for each reviewer, so that the sum of each reviewer's predicted ratings is 1. (Henceforth dubbed as **Norm**.)

Regardless of the chosen normalization scheme, we add the normalized predicted rating to the original preferences; unknown values in the original preference matrix are considered to be the mean rating value (2.5) to place them between the 'Ok' and 'Low' ratings. This forms our final input matrix **P**, which we feed into Taylor's optimization algorithm.

5.1 Topical relevance

To assess the topical relevance of the assignments, we evaluate them in terms of the mappings between papers/reviewers and subject categories. For every (paper,reviewer) assignment, we compute the dot product of the category vector of the paper with the category vector of the reviewer, and sum these dot products over the assignments made. Specifically paper-subject scores are recorded on a 2/1/0 scale (primary versus secondary versus neither) and reviewer-subject scores are recorded on a 1/-1/0 scale (interest versus conflict versus neither). In our dataset here, every paper has exactly one primary and one secondary category and hence the dot product can yield a number between -3 (reviewer has a conflict with both primary and secondary paper categories) and 3 (reviewer has interest in both paper categories). While other topical measures are certainly possible, the dot product method captures the relevance or 'on-topicness' of assignments made to each reviewer. We used a 90% training-10% test set split to learn our Norm and Resid models, and calculated the mean of the predicted ratings for each reviewer-paper pair across 100 iterations.

Fig. 2 depicts the results in terms of percentage improvement over the baseline Taylor approach (i.e., where only the original preferences without any additional data were input to the LP). Note that the topical evaluation metric shows a measurable improvement using our modified ratings **P** as input to Taylor's linear program. Since our new models take topical relevance into account, this is not unexpected. However, we accomplished this topical optimization without degrading the Taylor algorithm's original 'rating sum' objective; in fact, both the models considered here slightly improve this objective as well (see Fig. 2).

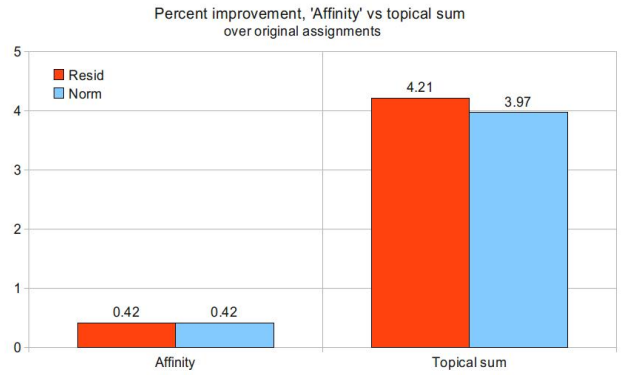

Figure 2: Topical relevance of assignments made with our approach versus Taylor's original formulation.

5.2 Assignment Quality

The common train-test split methodology, which was used in Section 3, is also useful for assessing assignment quality. Both prediction algorithm (8) and assignment algorithm (9) cannot see the given preferences within the test set. Clearly, the elimination of the test set's preferences limits the flexibility of the assignment algorithm, as it has a lower number of favorable preferences from which to choose. However, the prediction model fills this gap by providing estimates to all missing preferences, including those in the test set. This simulates the real life scenario, where the given reviewer ratings (corresponding to the training set) are limiting the possibilities of assignment algorithm, but by revealing more ratings to the algorithms (including the test set) they gain the flexibility to provide better assignments.

As the proportion of the test set increases, we take away more available preferences, which simulates an increasingly harsher assignment environment. Accordingly, we evaluated several possible proportions, ranging from 50% of the given preferences within the test set, to 30% of preferences in the test set. In each experiment, we employed a series of 20 random train-test split and evaluated assignment quality. The baseline is Taylor's original algorithm, where all missing ratings, including those in the test set, are treated as "unknowns." We compare this baseline against the two aforementioned alternatives, Resid and Norm.

We evaluate quality of assignments by their ability to make good use of the hidden ratings in the test set. The results are presented in Figs. 3, 4, & 5, and were fairly consistent over the different proportions of the test set. As illustrated here, the predominant number (around 60-70%) of test assignmentsmade using the original preference matrix fall in the unpreferred ("No") category. On the other hand, when imputing the missing ratings, using either Resid or Norm, the balance completely changes in favor of higher quality preferences. Resid makes about 50-60% of test assignments out of the highest quality ratings ("High"), and only about 15% of test assignments are bad ("No"). Norm is close, but not quite as good as Resid, a difference that should be further investigated over additional datasets. Overall we find the results strongly support our goal to increase assignment quality by providing more flexibility with additional ratings from which to choose.

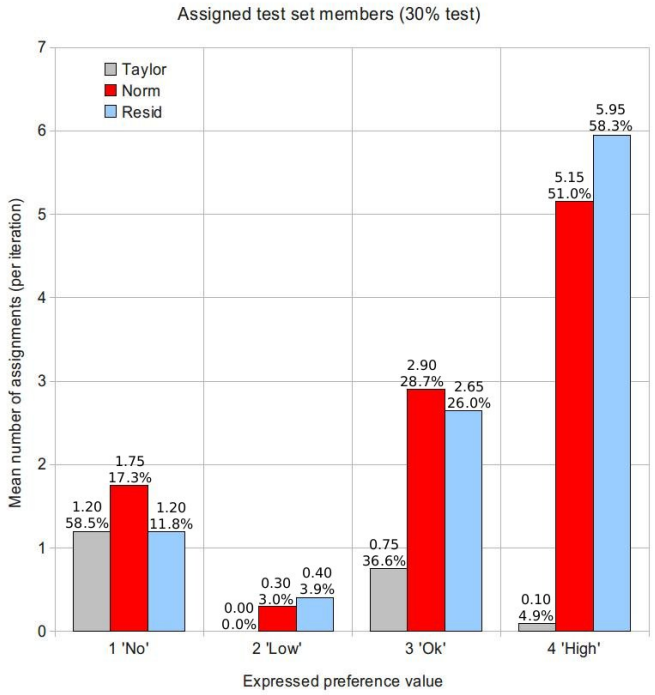

Figure 3: Evaluating the assignments made by the unmodified Taylor algorithm and the new preference models w.r.t. reviewers' four categories of preferences, using a 70-30 test-training set split, averaged across 20 iterations. Mean assignments per iteration, and each value's percent of assignments for each iteration, are indicated above each bar.

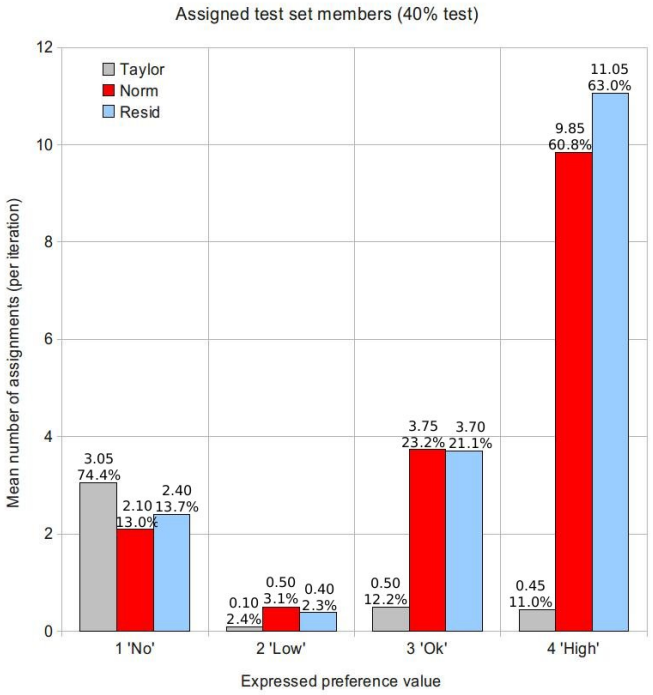

Figure 4: Evaluating the assignments made by the unmodified Taylor algorithm and the new preference models, using a 60-40 test-training set split.

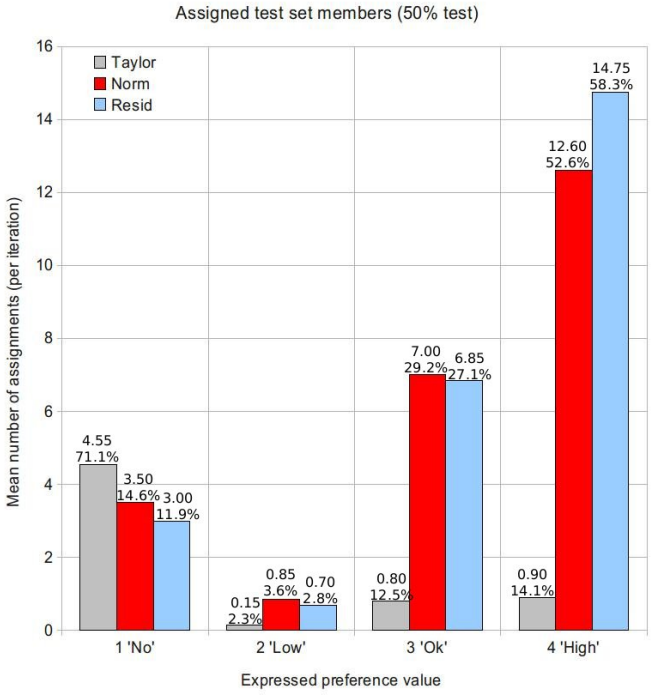

Figure 5: Evaluating the assignments made by the unmodified Taylor algorithm and the new preference models, using a 50-50 test-training set split.

6. DISCUSSION

We have investigated the modeling of paper-reviewer preferences within a conference management system. The very limited data, typical to this context, requires identifying and exploiting multiple sources of information within a hybrid recommendation model. The proposed models provide improved predictions of reviewer preferences. More importantly, we showed how the improved modeling of such preferences can lead to improvements in actual review assignments. Encouraging experimental results demonstrate that the improved modeling can be well worth the effort in ensuring satisfaction of conference program committee reviewers. A key question for future work is to provide theoretical justification for the empirical evidence presented here. We also intend to field the recommendation capabilities presented here in a real conference management system and gain further insights into the issues involved.

Acknowledgements

The authors acknowledge the approval of Prof. Xindong Wu, Steering Committee chair of the IEEE ICDM conference series for the use of preference/bid data collected during the ICDM'07 conference reviewing process, and associated information about papers/reviewers. All datasets were suitably anonymized before the modeling and analysis steps conducted here.

7. REFERENCES

[1] C. Basu, H. Hirsh, W. Cohen, and C. Nevill-Manning. Technical paper recommendation: a study in combining multiple information sources. *Journal of AI Research*, pages 231–252, 2001.

[2] S. Benferhat. Conference paper assignment. *International Journal of Intelligent Systems*, 16(10):1183, 2001.

[3] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] J. Canny. Collaborative filtering with privacy via factor analysis. In *Proc. SIGIR'02*, pages 238–245, 2002.

[5] S. T. Dumais and J. Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *Proc. SIGIR '92*, pages 233–244, 1992.

[6] D. Goldberg, D. Nichols, B. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. of the ACM*, 35:61–70, 1992.

[7] J. Goldsmith and R. H. Sloan. The AI conference paper assignment problem. In *Pref. Handling for AI, Papers from the AAAI Workshop*, 2007.

[8] D. Hartvigsen, J. C. Wei, and R. Czuchlewski. The conference paper-reviewer assignment problem. *Decision Sciences*, 30(3):865–876, 1999.

[9] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Info. Systems*, 22:89–115, 2004.

[10] J. E. Hopcroft and R. M. Karp. An $n^{2.5}$ algorithm for maximum matching in bipartite graphs. *SIAM Journal on Computing*, 18:225–231, 1973.

[11] J. A. Konstan, B. N. Miller, D. Maltz, J. L. Herlocker, L. R. Gordon, and J. Riedl. GroupLens: applying collaborative filtering to usenet news. *Commun. of the ACM*, 40(3):77–87, 1997.

[12] Y. Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. KDD'08*, pages 426–434, 2008.

[13] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[14] N. D. Mauro, T. M. A. Basile, and S. Ferilli. GRAPE: an expert review assignment component for scientific conference management systems. In *Proc. IEA/AIE'2005*, pages 789–798, 2005.

[15] S. McNee, J. Riedl, and J. Konstan. Being accurate is not enough: how accuracy metrics have hurt recommender systems. In *CHI Extended Abstracts*, pages 1097–11101, 2006.

[16] D. Mimno and A. McCallum. Expertise modeling for matching papers with reviewers. In *Proc. KDD'07*, pages 500–509, 2007.

[17] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *Proc. KDD Cup and Workshop*, 2007.

[18] R. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proc. of the 17th Conf. on Uncertainty in AI*, pages 437–444, 2001.

[19] N. Ramakrishnan, O. Zaiaine, Y. Shi, C. Clifton, and X. Wu. Proc. ICDM'07, 2007.

[20] P. Rigaux. An iterative rating method: application to web-based conference management. In *Proc. SAC'04*, pages 1682–1687, 2004.

[21] R. Salakhutdinov and Mnih. Probabilistic matrix factorization. In *Proc. NIPS'07*, pages 1257–1264, 2008.

[22] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted boltzmann machines for collaborative filtering. In *Proc. 24th Annual Intl. Conf. on Machine Learning*, pages 791–798, 2007.

[23] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *Proc. 10th Intl. Conf. on the World Wide Web*, pages 285–295, 2001.

[24] G. Takacs, I. Pilaszy, B. Nemeth, and D. Tikk. Major components of the gravity recommendation system. *SIGKDD Explorations*, 9:80–84, 2007.

[25] C. J. Taylor. On the optimal assignment of conference papers to reviewers. Technical Report MS-CIS-08-30, University of Pennsylvania, 2008.

[26] X. Wei and W. B. Croft. LDA-based document models for ad-hoc retrieval. In *Proc. SIGIR '06*, pages 178–185, 2006.

[27] D. Yarowsky and R. Florian. Taking the load off the conference chairs: towards a digital paper-routing assistant. In *Proc. EMNLP'99.*, 1999.